

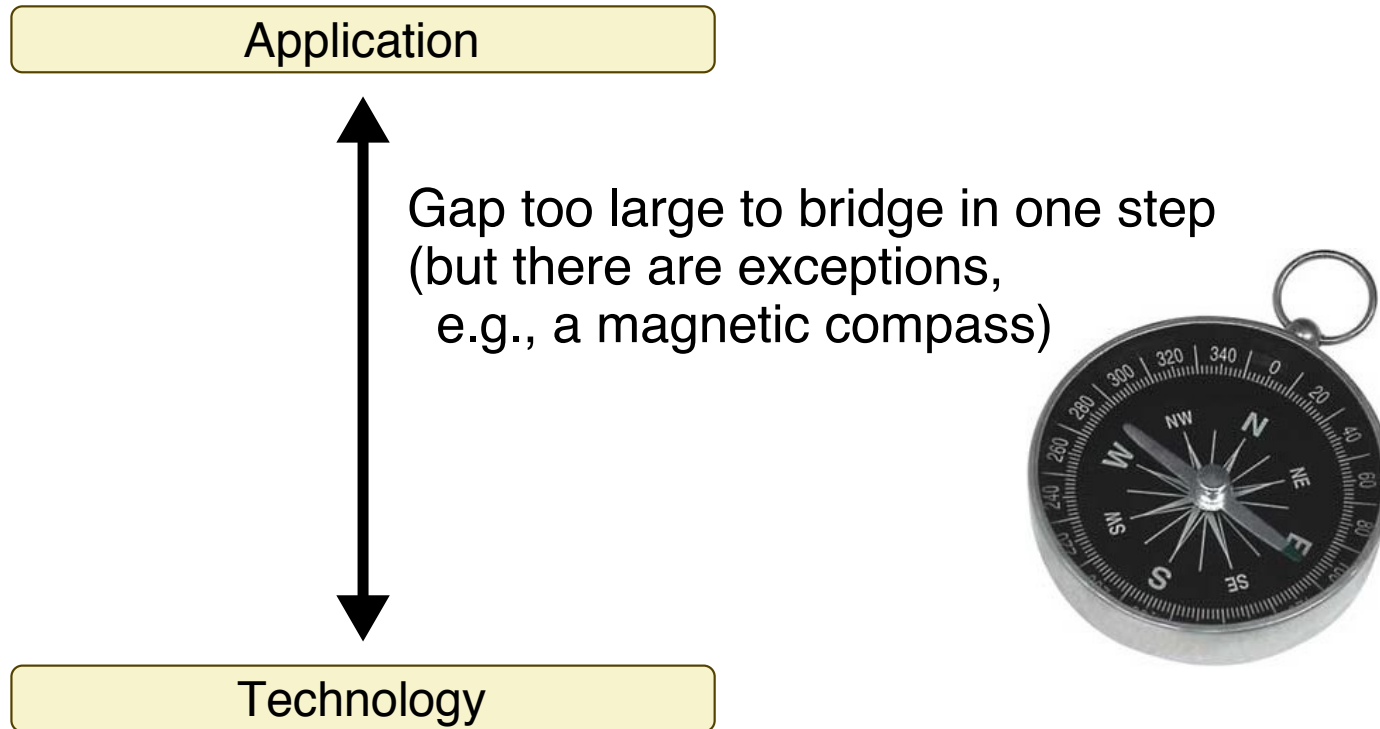
ECE 4750 Computer Architecture Course Overview

Christina Delimitrou

School of Electrical and Computer Engineering
Cornell University

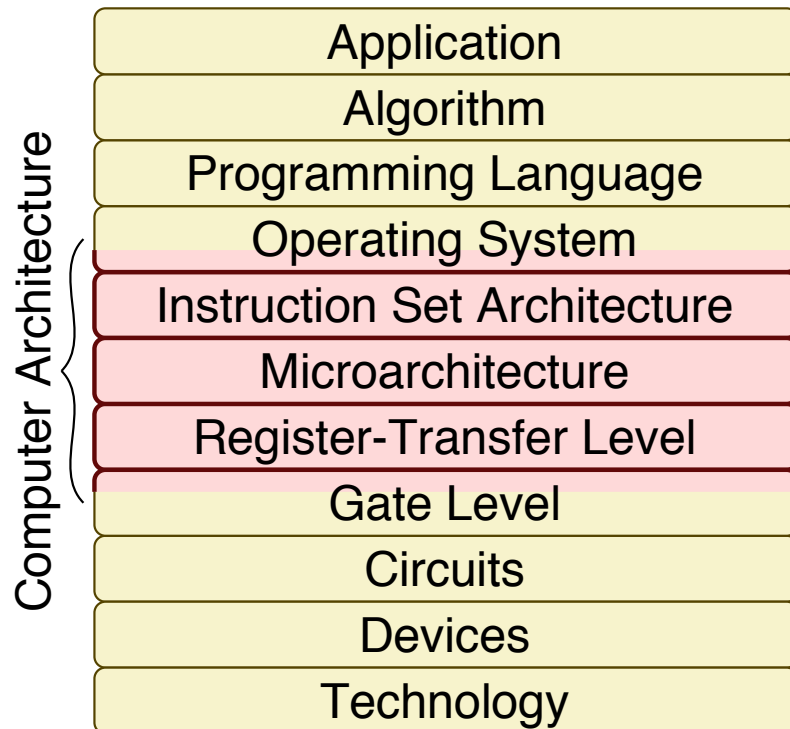
<http://www.csl.cornell.edu/courses/ece4750>

The Computer Systems Stack



In its broadest definition, computer architecture is the **development of the abstraction/implementation layers** that allow us to execute information processing **applications** efficiently using available manufacturing **technologies**

The Computer Systems Stack



Sort an array of numbers

2,6,3,8,4,5 -> 2,3,4,5,6,8

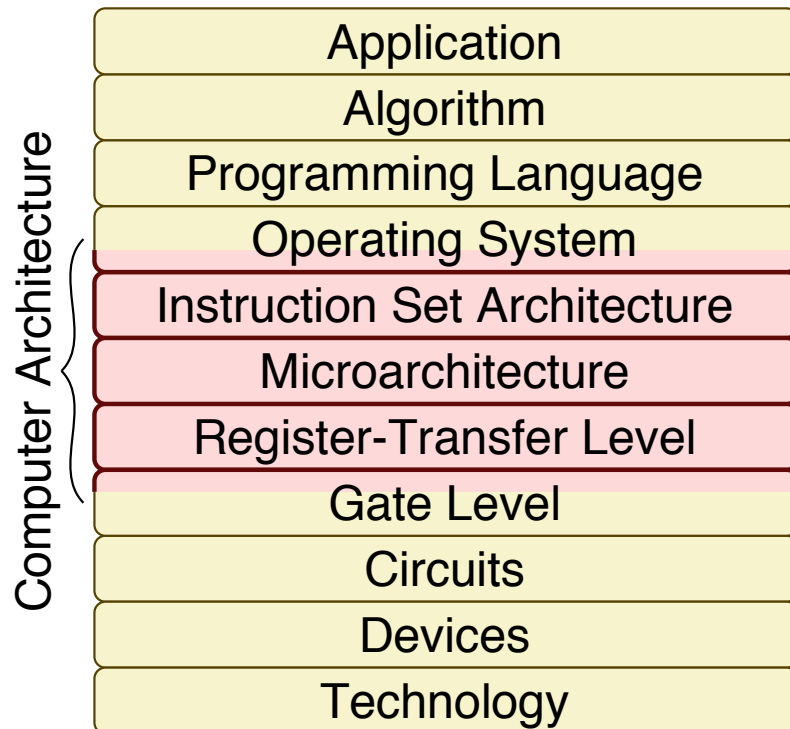
Out-of-place selection sort algorithm

1. Find minimum number in array
2. Move minimum number into output array
3. Repeat steps 1 and 2 until finished

C implementation of selection sort

```
void sort( int b[], int a[], int n ) {  
    for ( int idx, k = 0; k < n; k++ ) {  
        int min = 100;  
        for ( int i = 0; i < n; i++ ) {  
            if ( a[i] < min ) {  
                min = a[i];  
                idx = i;  
            }  
        }  
        b[k] = min;  
        a[idx] = 100;  
    }  
}
```

The Computer Systems Stack



Mac OS X, Windows, Linux

Handles low-level hardware management

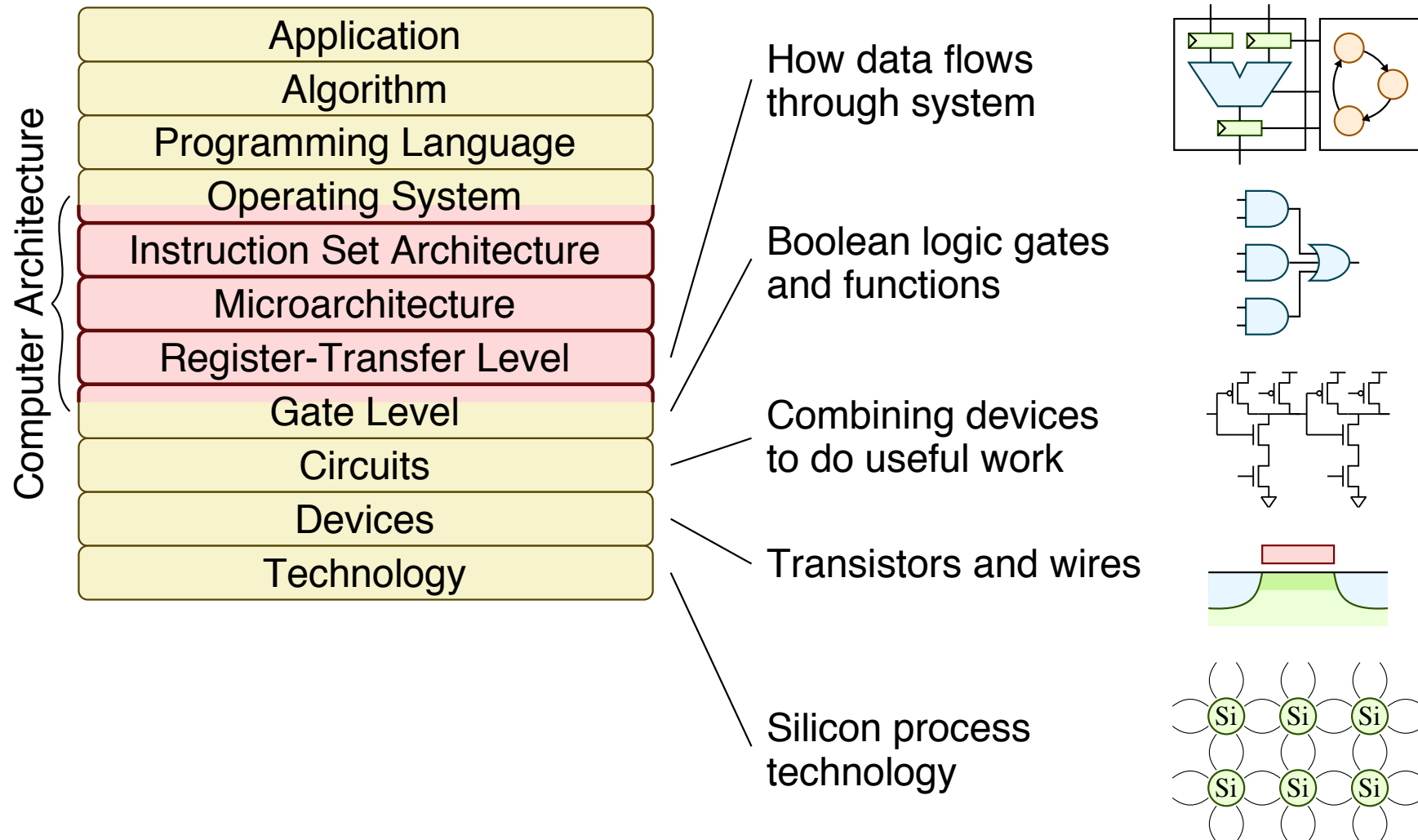


MIPS32 Instruction Set

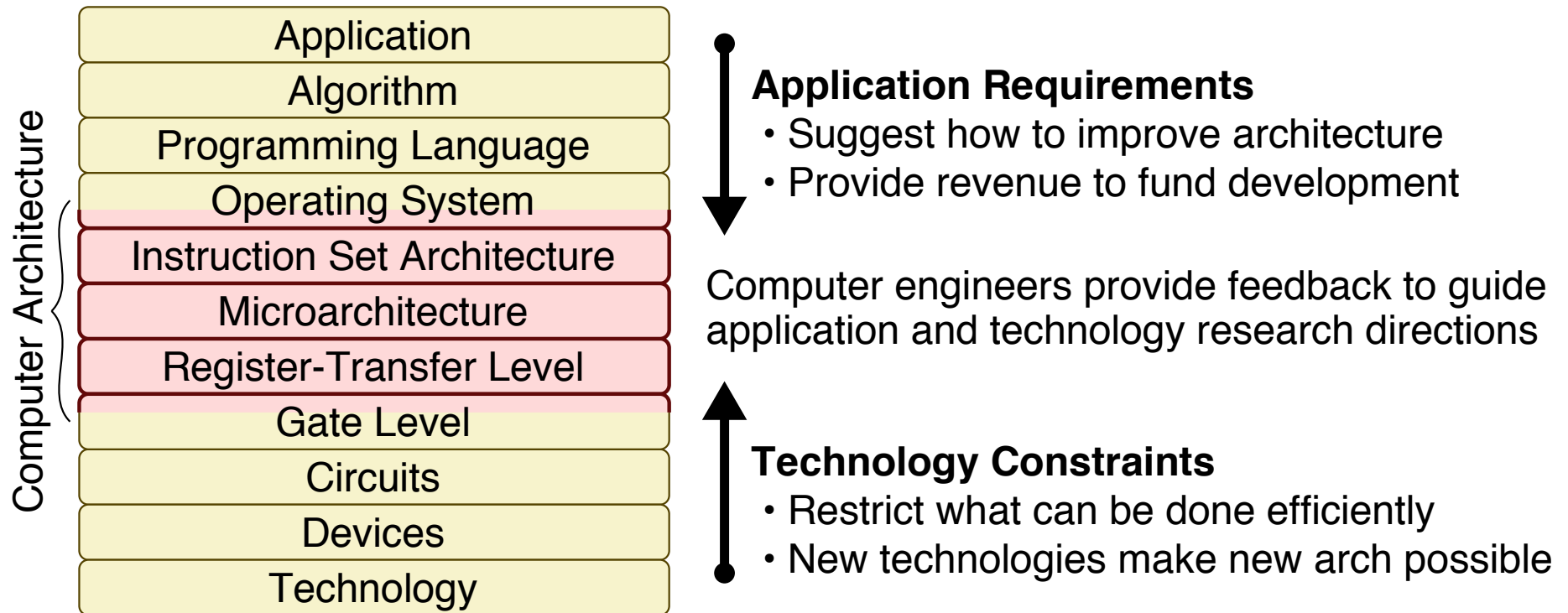
Instructions that machine executes

```
blez    $a2, done
move    $a7, $zero
li      $t4, 99
move    $a4, $a1
move    $v1, $zero
li      $a3, 99
lw      $a5, 0($a4)
addiu   $a4, $a4, 4
slt     $a6, $a5, $a3
movn    $v0, $v1, $a6
addiu   $v1, $v1, 1
movn    $a3, $a5, $a6
```

The Computer Systems Stack

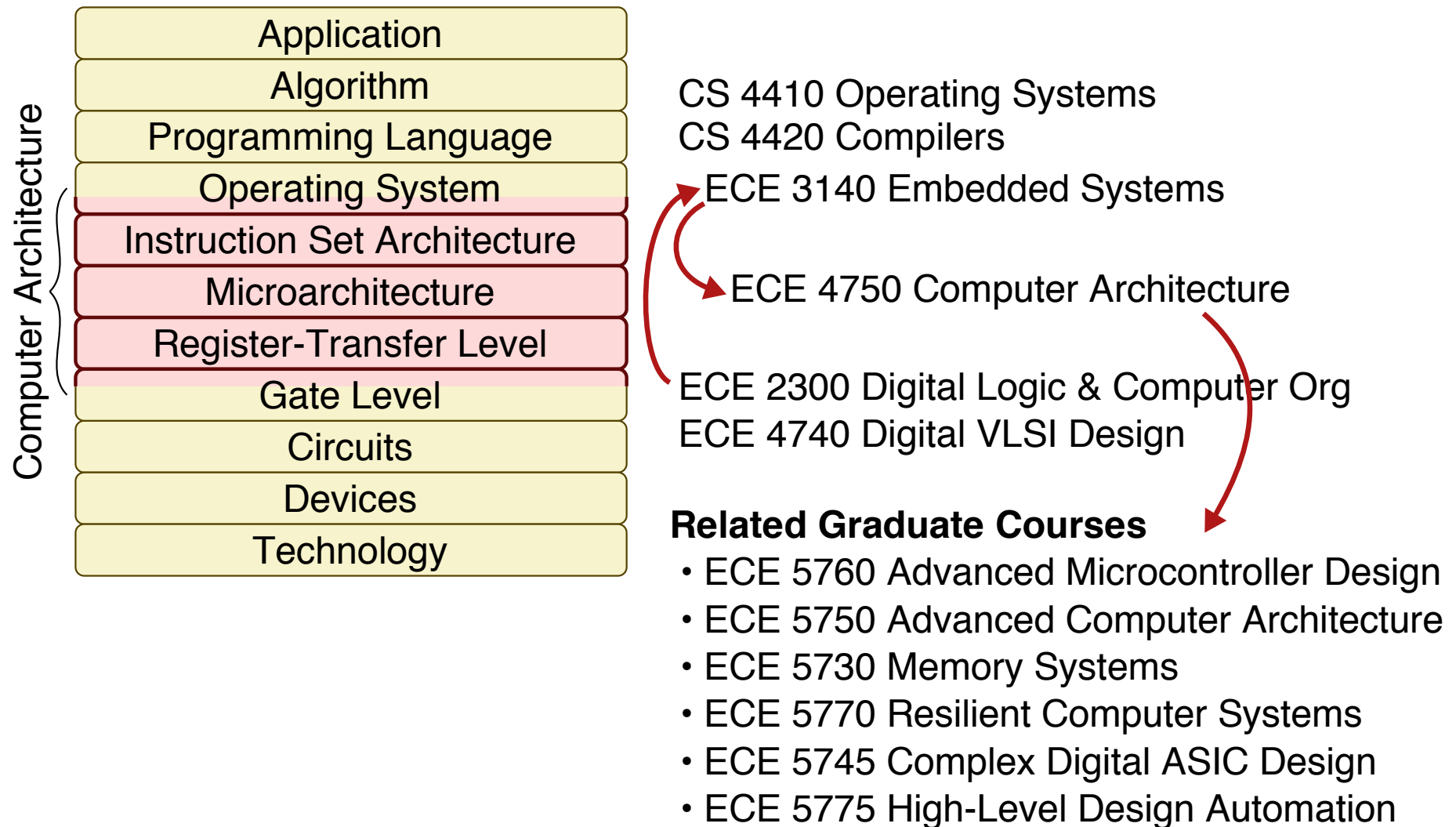


Application Requirements vs. Technology Constraints

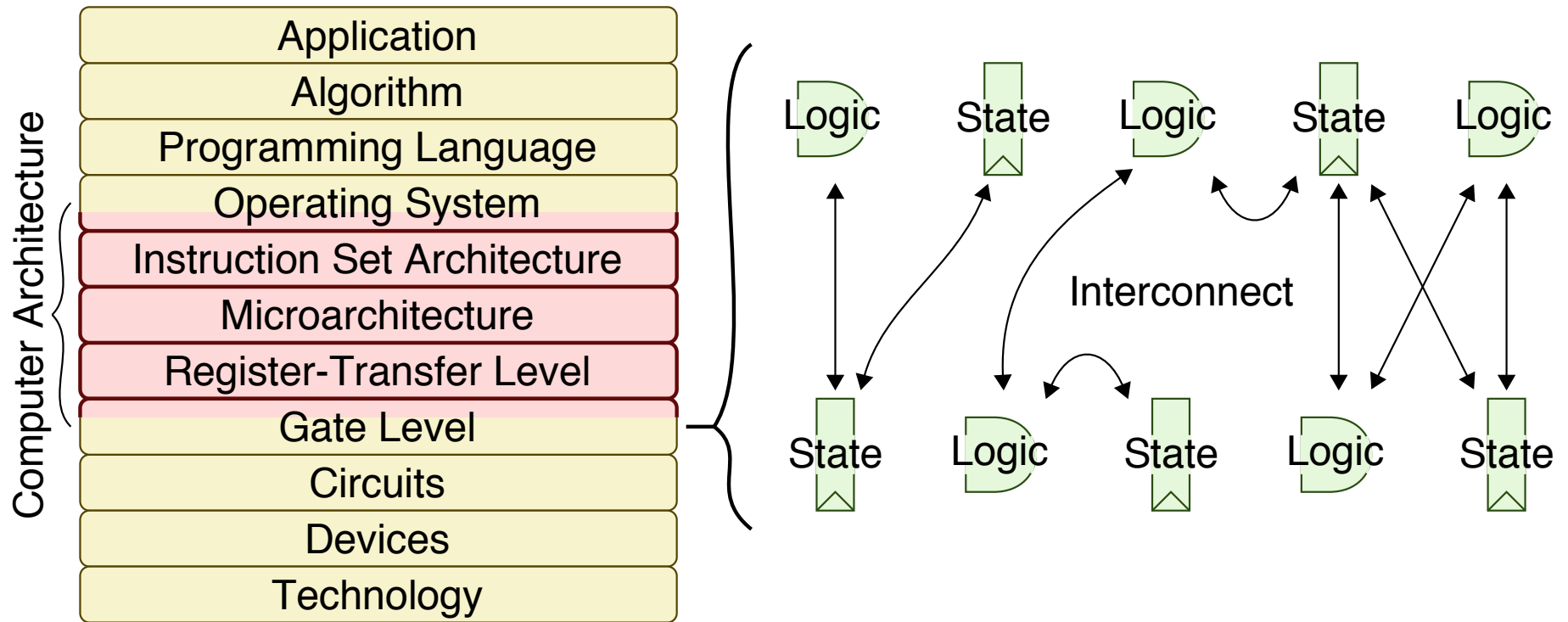


In its broadest definition, computer architecture is the **development of the abstraction/implementation layers** that allow us to execute information processing **applications** efficiently using available manufacturing **technologies**

Computer Architecture in the ECE/CS Curriculum



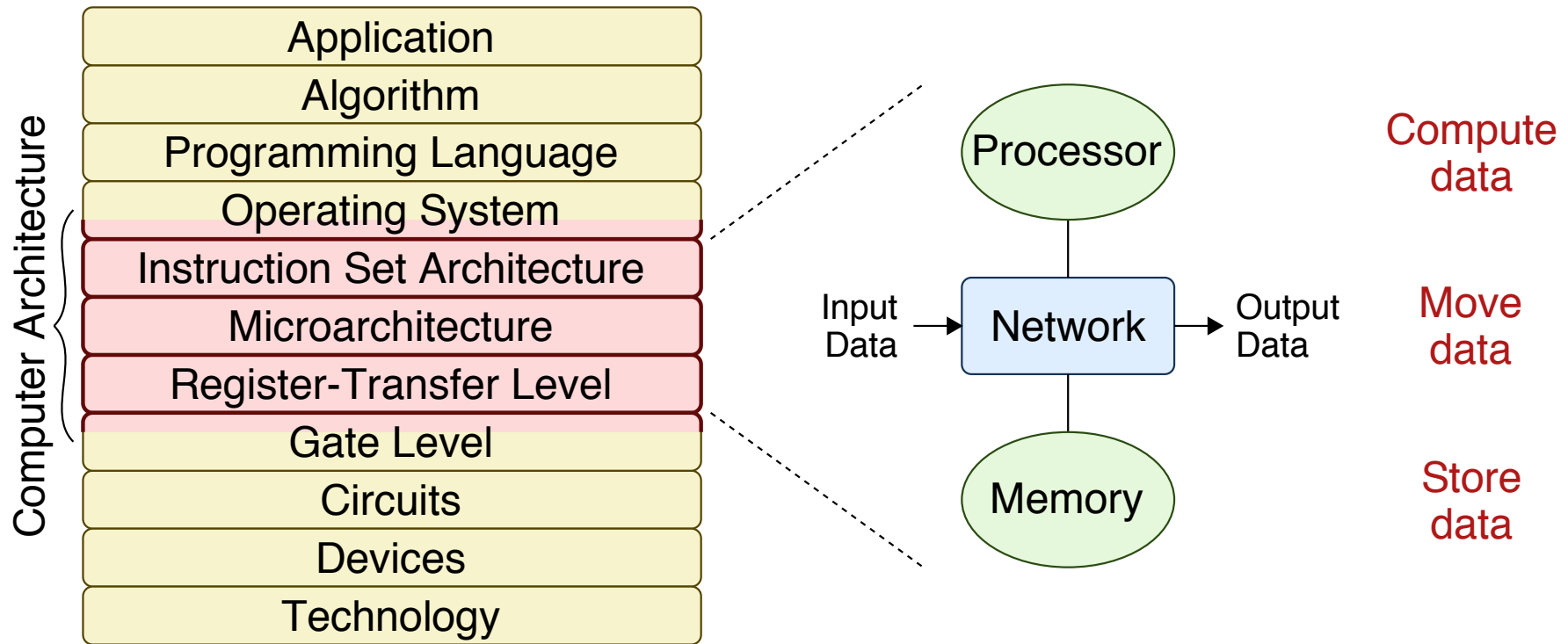
Logic, State, and Interconnect



Digital systems are implemented with three basic building blocks

- **Logic** to process data
- **State** to store data
- **Interconnect** to move data

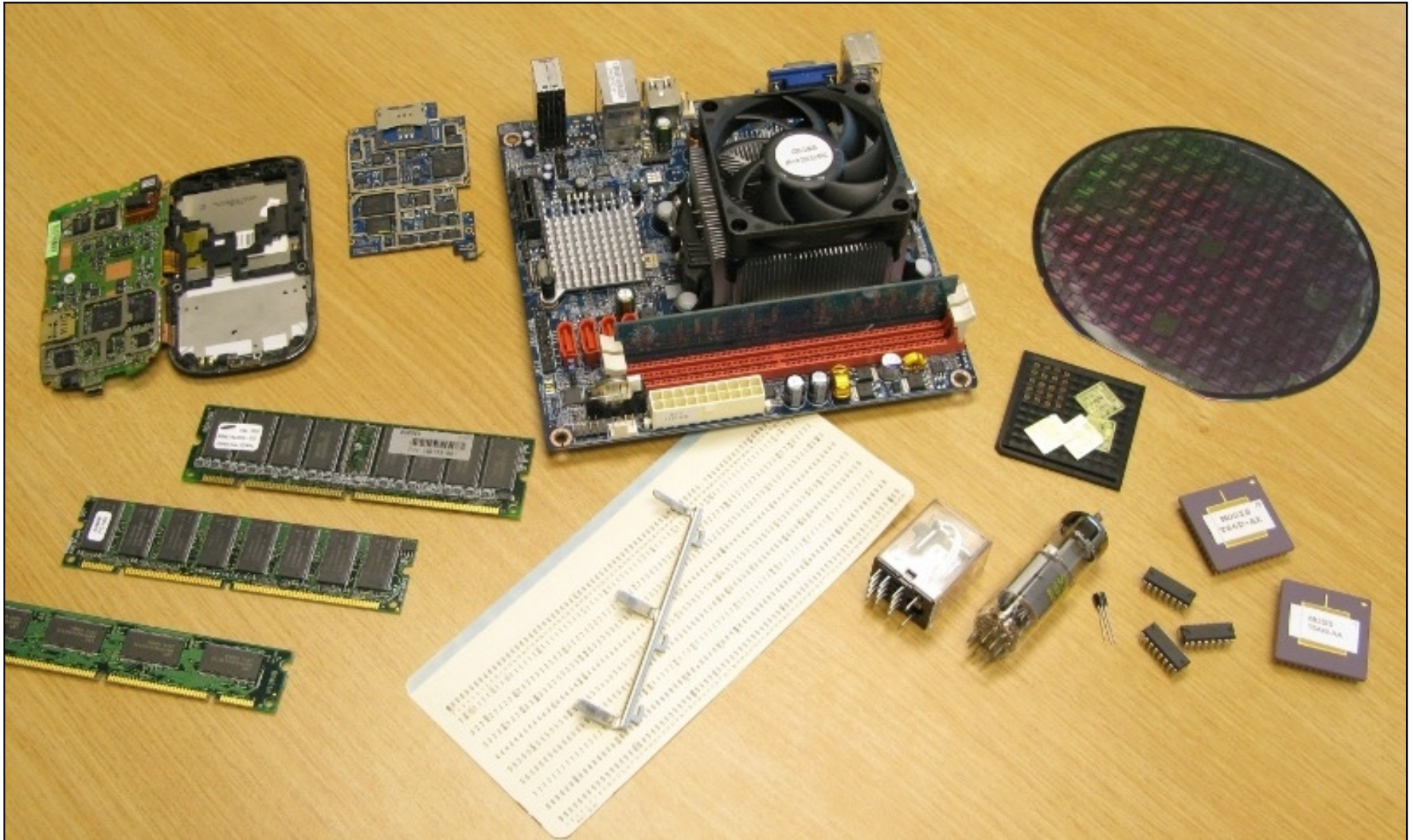
Processors, Memories, and Networks



Computer engineering basic building blocks

- **Processors** for computation
- **Memories** for storage
- **Networks** for communication

Computer Architecture Artifacts



Application

Algorithm

PL

OS

ISA

 μ Arch

RTL

Gates

Circuits

Devices

Technology

Agenda

What is Computer Architecture?

Activity 1

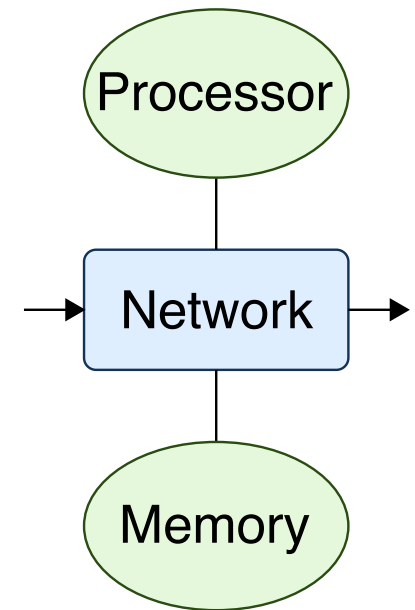
Trends in Computer Architecture

Activity 2

Computer Architecture Design

Activity #1: Sorting with a Sequential Processor

- ▶ **Application:** Sort 32 numbers
- ▶ **Simulated Sequential Computing System**
 - ▷ Processor: You!
 - ▷ Memory: Worksheet, read input data, write output data
 - ▷ Network: Passing/collecting the worksheets
- ▶ **Activity Steps**
 - ▷ 1. Discuss strategy with neighbors
 - ▷ 2. When instructor starts timer, flip over worksheet
 - ▷ 3. Sort 32 numbers as fast as possible
 - ▷ 4. Lookup when completed and write time on worksheet
 - ▷ 5. Raise hand
 - ▷ 6. When everyone is finished, then analyze data



Application

Algorithm

PL

OS

ISA

 μ Arch

RTL

Gates

Circuits

Devices

Technology

Agenda

What is Computer Architecture?

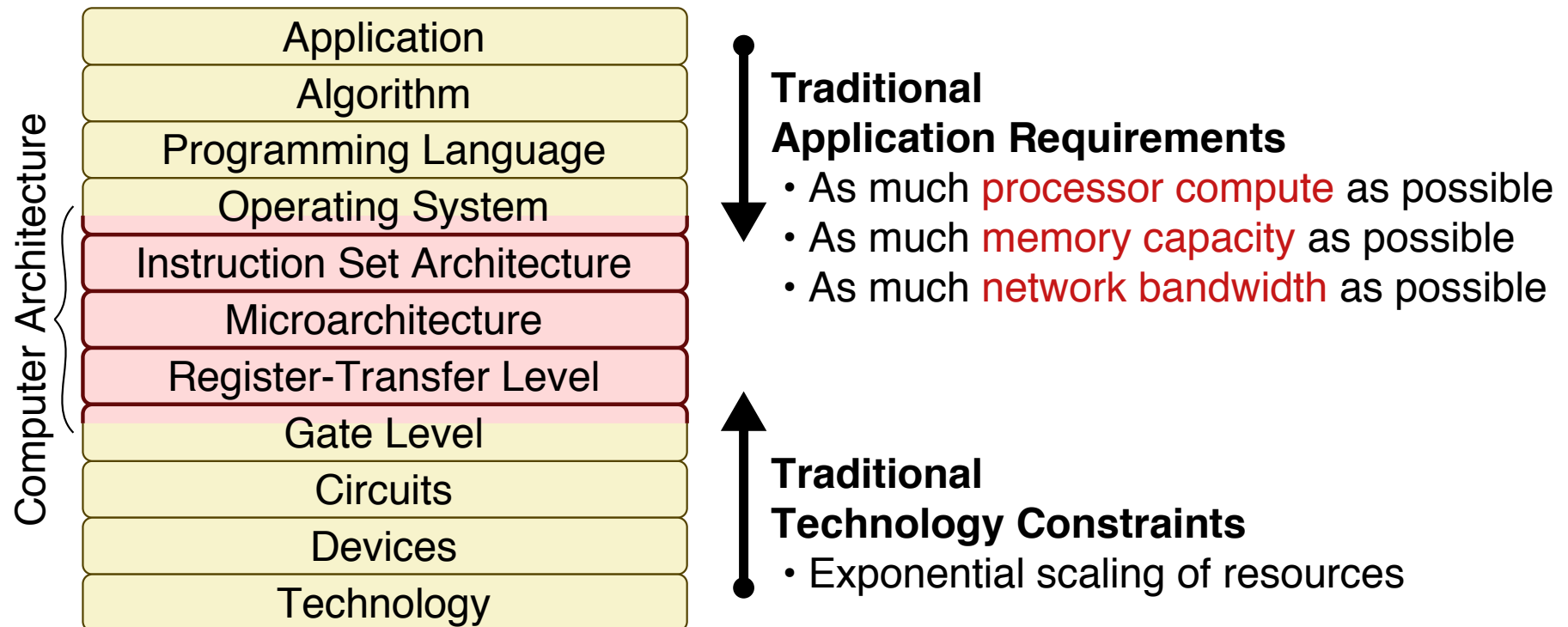
Activity 1

Trends in Computer Architecture

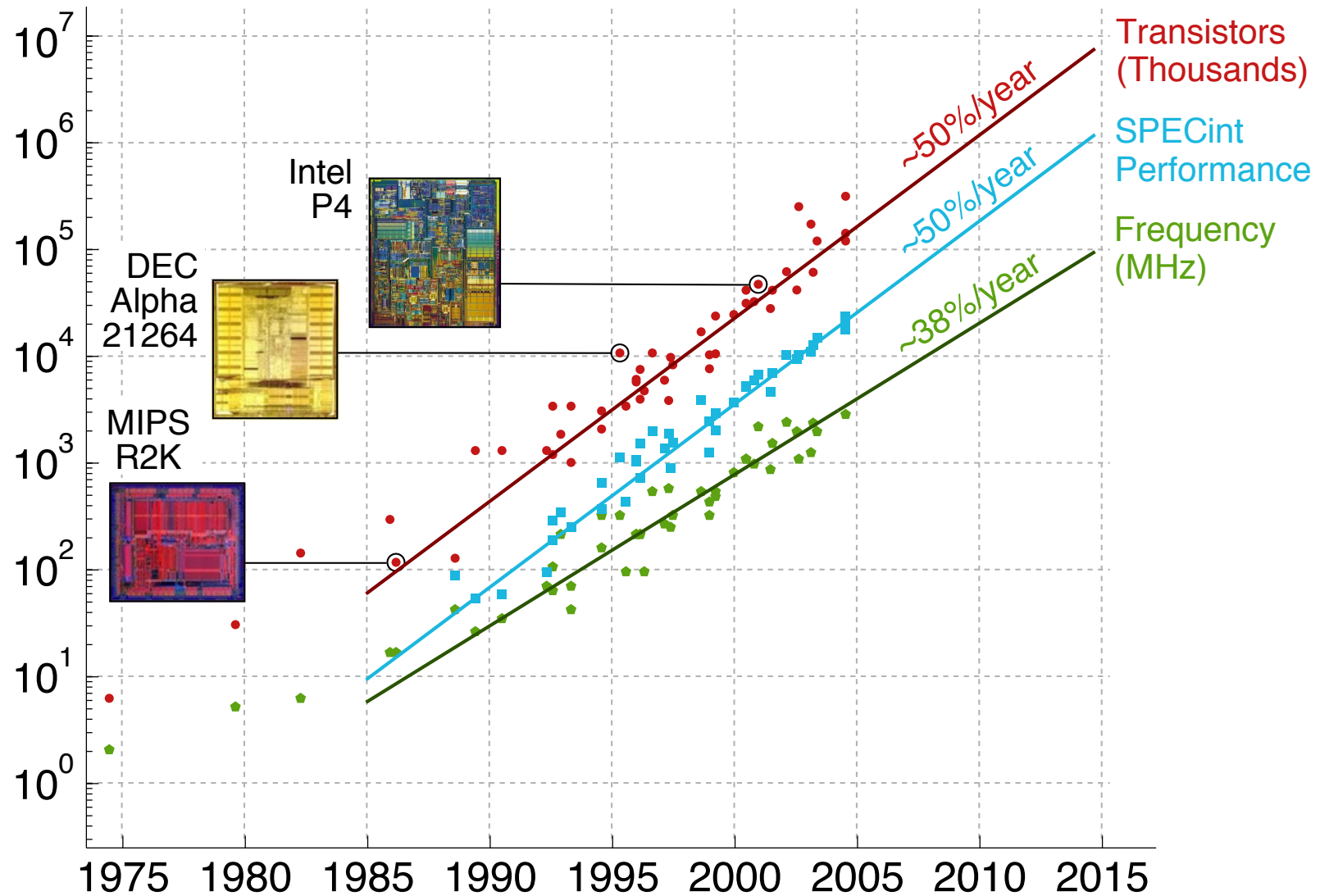
Activity 2

Computer Architecture Design

Application Requirements vs. Technology Constraints

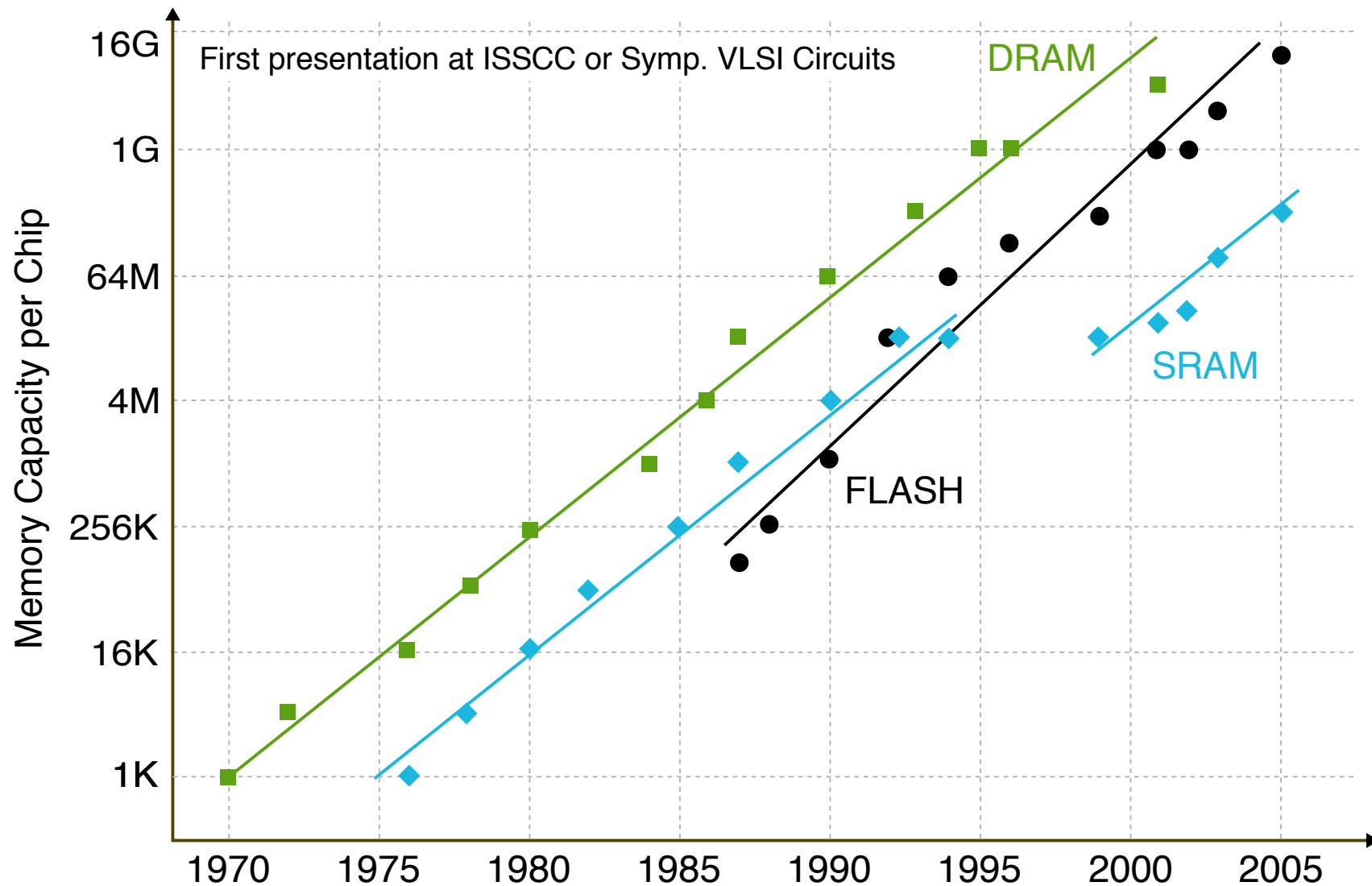


Exponential Scaling for Processor Computation



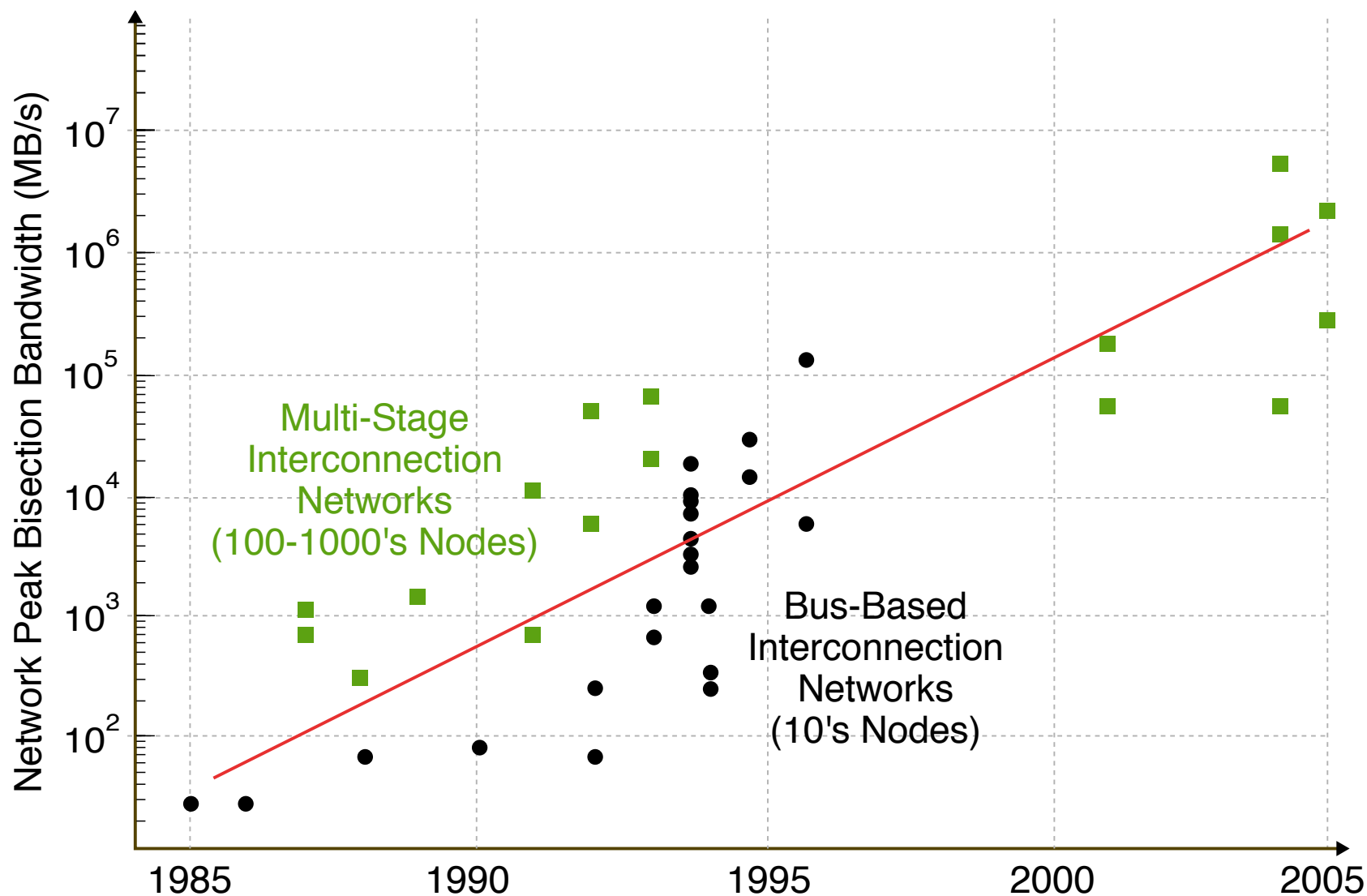
Data collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, C. Batten

Exponential Scaling for Memory Capacity



Adapted from K. Itoh et al. "Ultra-Low Voltage Nano-Scale Memories." Spring 2007.

Exponential Scaling for Network Bandwidth



Data from Hennessy & Patterson, Morgan Kaufmann, 2nd & 5th eds., 1996 & 2011; D.E. Culler et al., Morgan Kaufmann, 1999.

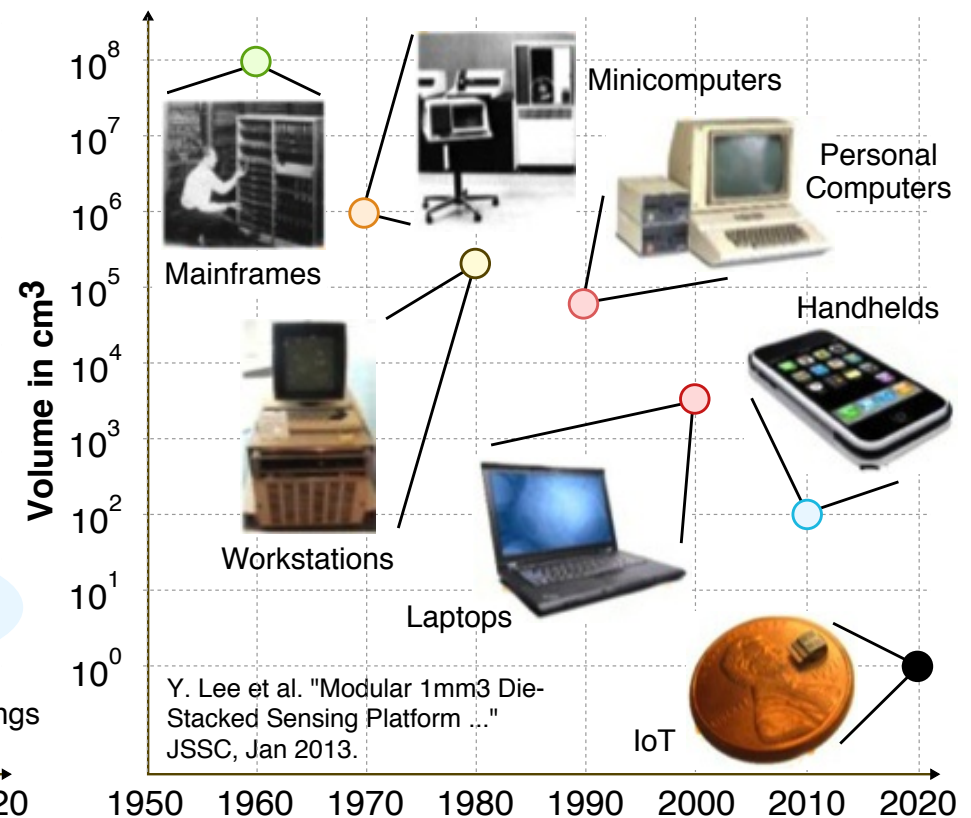
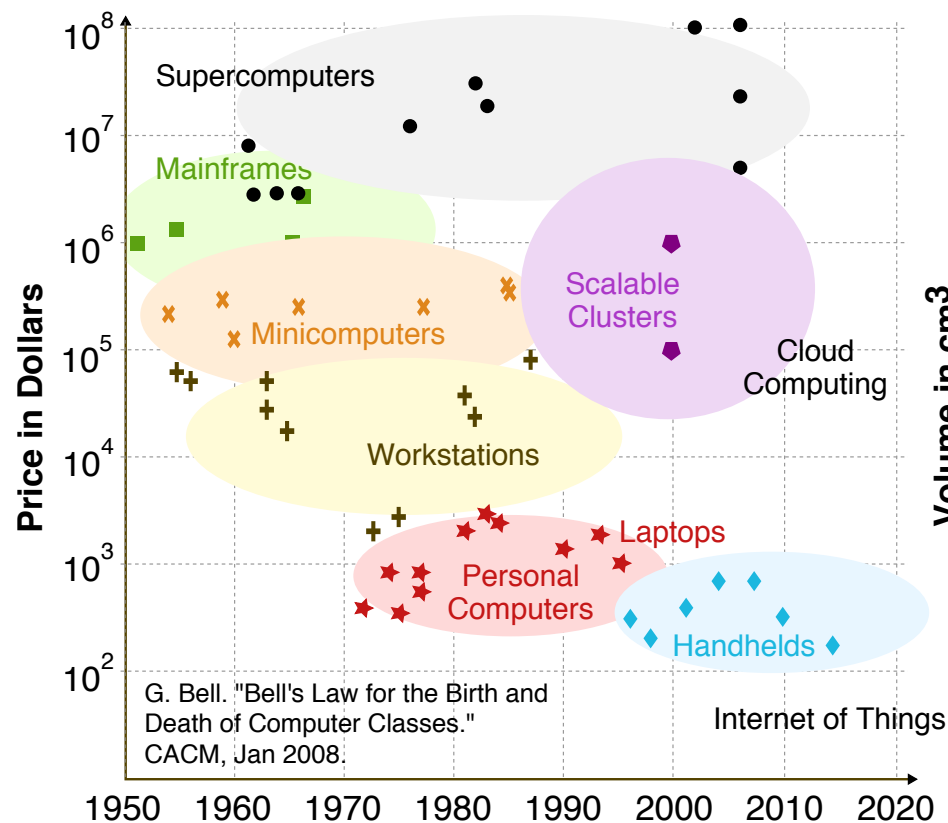
Key trends in application requirements and technology constraints over the past decade have resulted in a radical rethinking of the processors, memories, and networks used in modern computing systems

Five Key Trends in Computer Architecture

1. Growing diversity in application requirements motivate growing diversity in computing systems pushing towards the cloud and IoT
2. Energy & power constrain systems across the computing spectrum
3. Transition to multiple cores integrated onto a single chip
4. Transition to heterogeneous systems-on-chip
5. Technology scaling challenges motivate new emerging compute, storage, and communication device technologies

Trend 1: Bell's Law

Roughly every decade a new, smaller, lower priced computer class forms based on a new programming platform resulting in entire new industries



Trend 1: Growing Diversity in Apps & Systems



Trend 2: Energy and Power Constraints



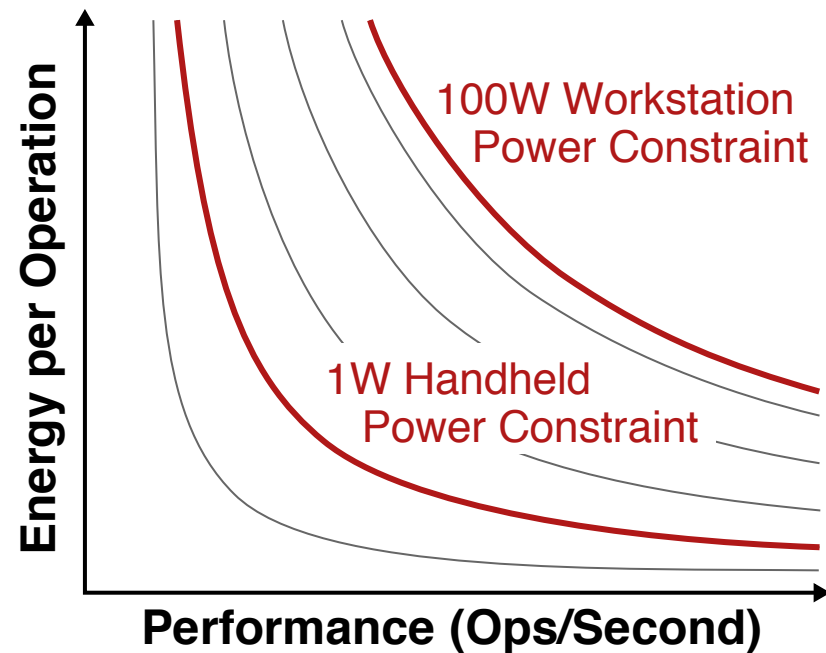
$$\text{Power} = \frac{\text{Energy}}{\text{Second}} = \frac{\text{Energy}}{\text{Op}} \times \frac{\text{Ops}}{\text{Second}}$$

Power

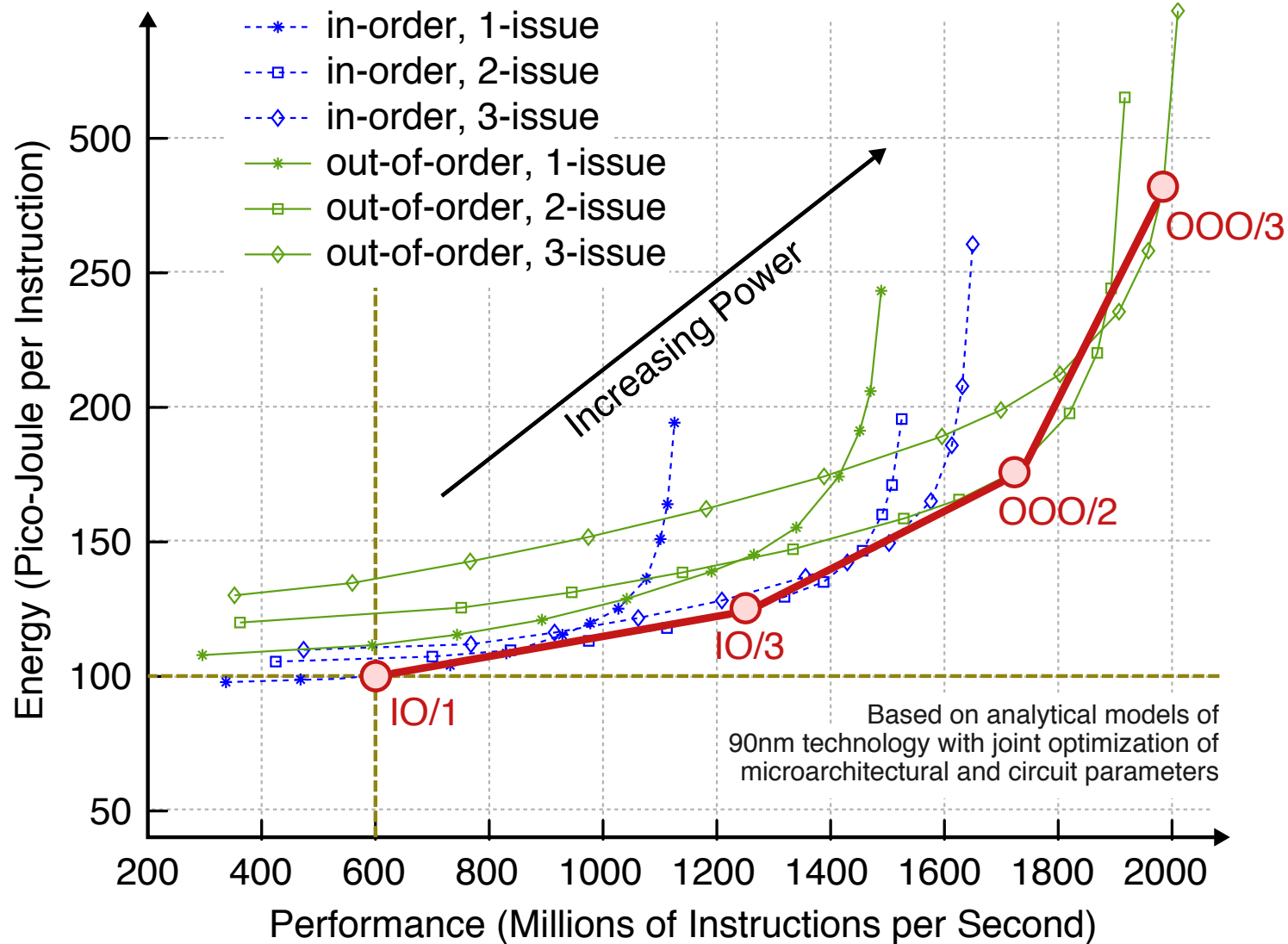
Chip Packaging
Chip Cooling
System Noise
Case Temperature
Data-Center Air
Conditioning

Energy

Battery Life
Electricity Bill
Mobile Device
Weight

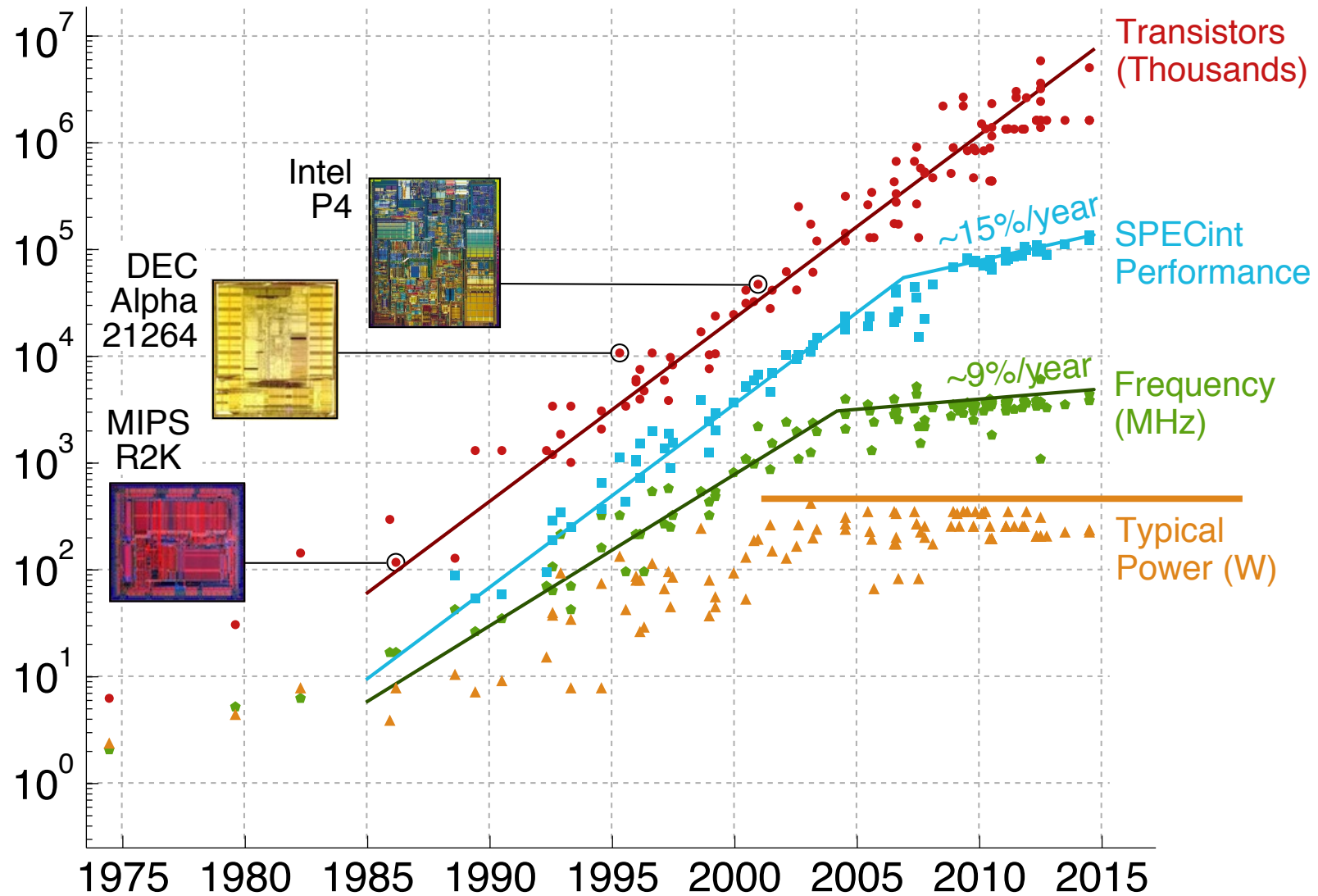


Trend 2: Energy and Performance of Single Processor



Adpated from O. Azizi et al. "Energy-Performance Tradeoffs ..." ISCA, 2010.

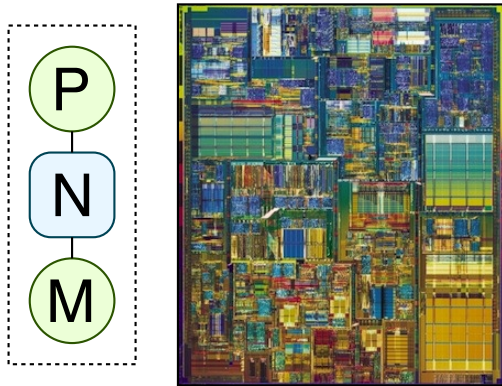
Trend 2: Power Constrains Single-Processor Scaling



Trend 3: Transition to Multicore Processors

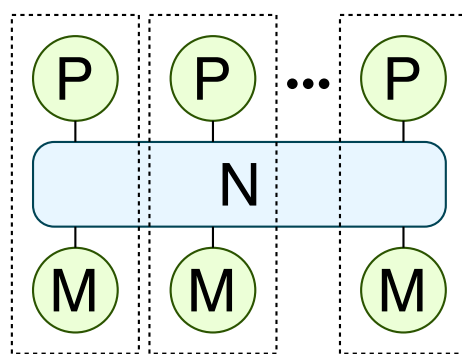
Intel Pentium 4

Single monolithic processor



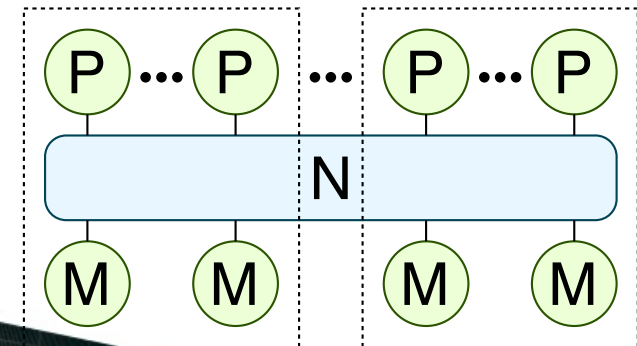
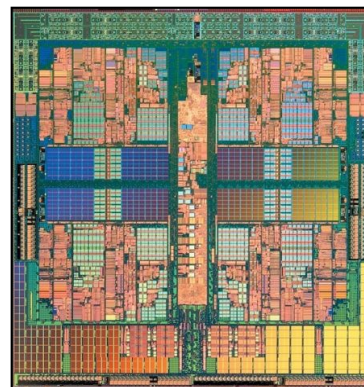
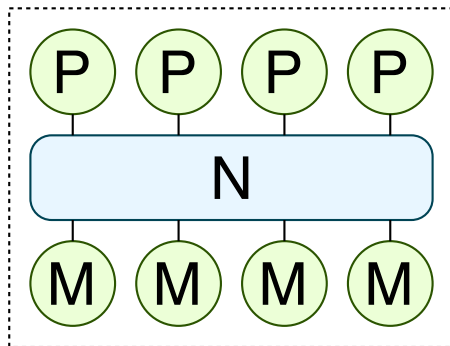
Cray XT3 Supercomputer

1024 single-core processors



AMD Quad-Core Opteron

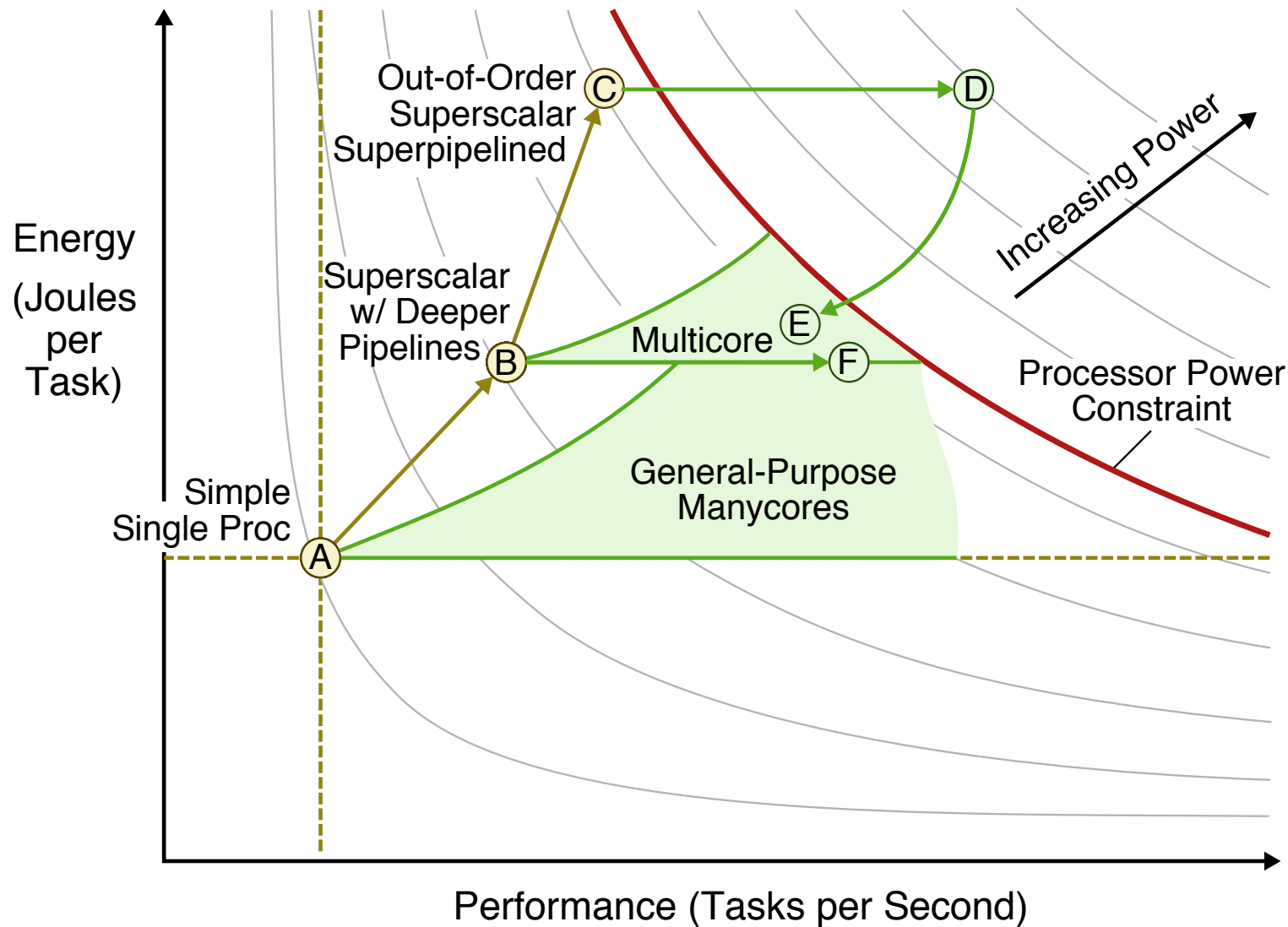
Four cores on the same die



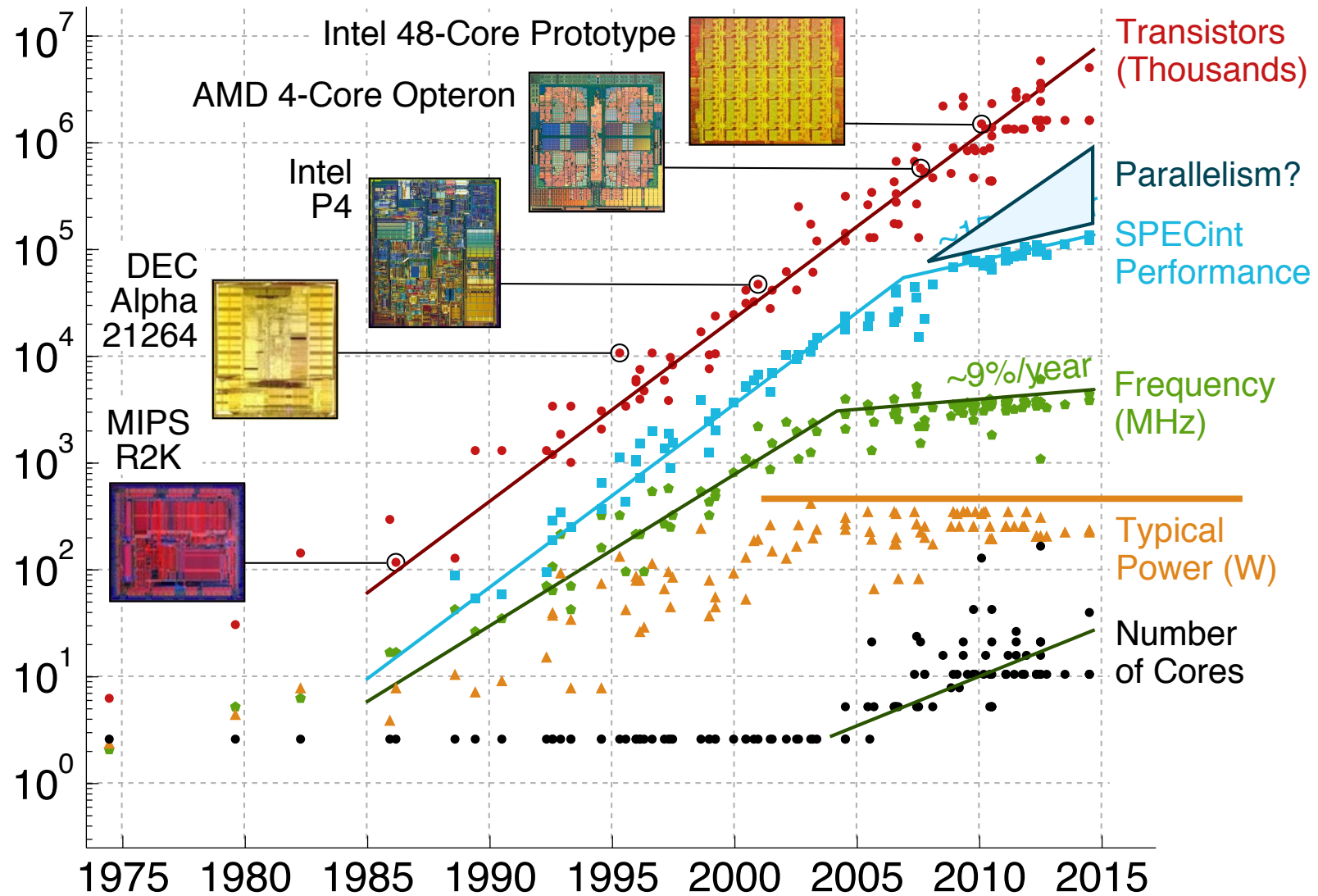
IBM Blue Gene Q Supercomputer

Thousands of 18-core processors

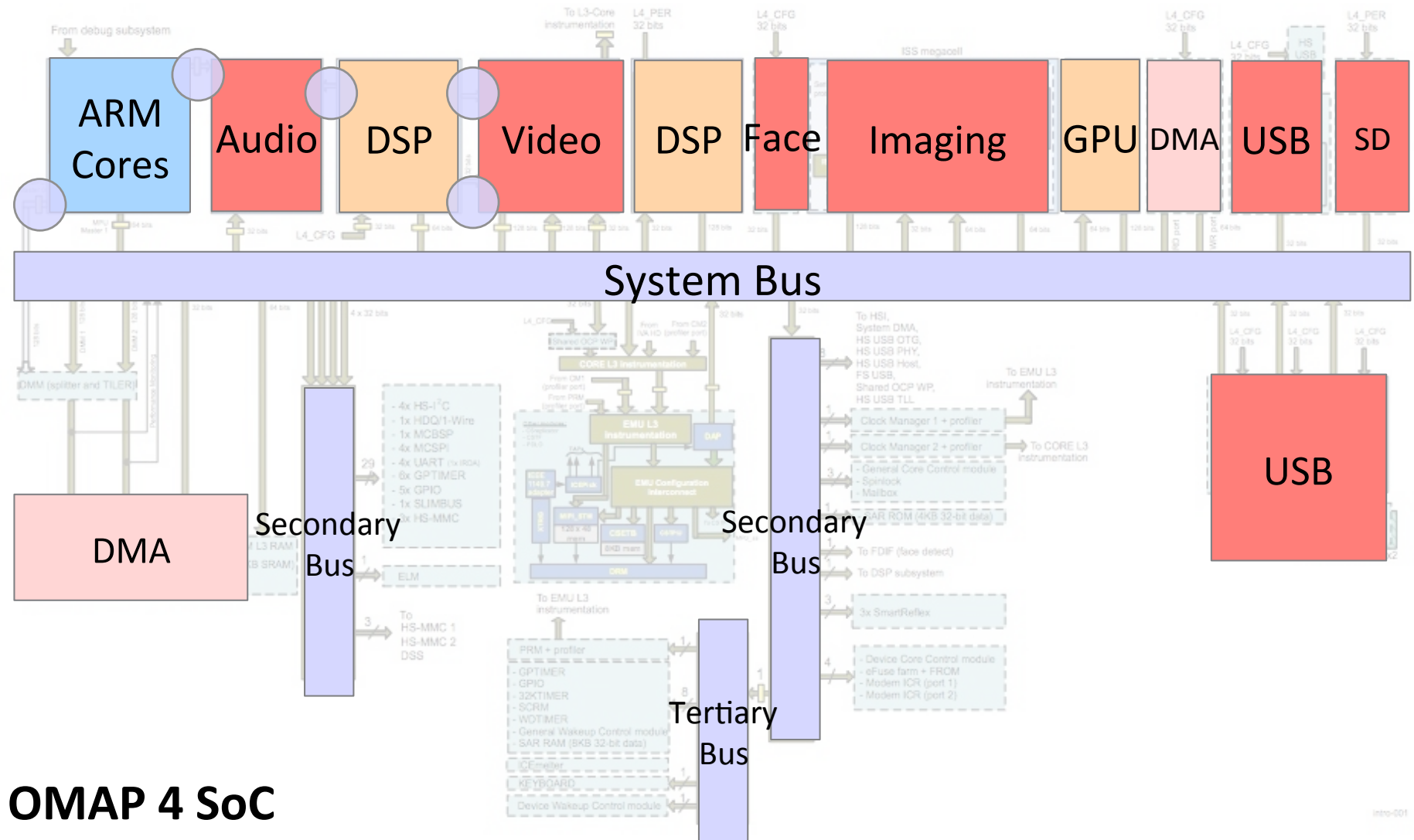
Trend 3: Energy and Performance of Multicores



Trend 3: The Multicore “Hail Mary Pass”

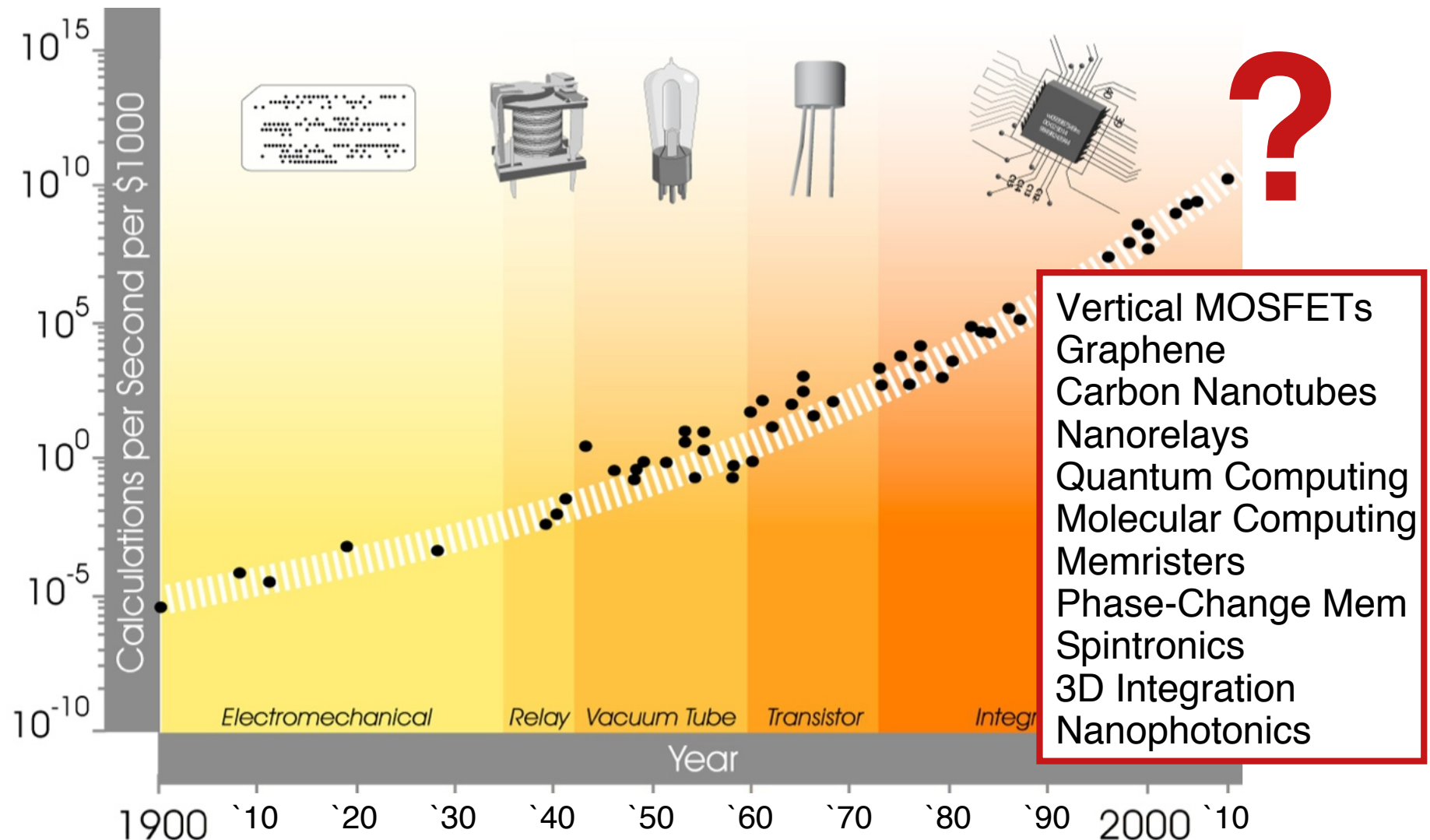


Trend 4: Heterogeneous Systems-on-Chip



Adapted from D. Brooks Keynote at NSF XPS Workshop, May 2015.

Trend 5: Emerging Device Technologies



Adapted from R. Kurzweil. "The Singularity is Near." Penguin Books, 2006.

Key trends in application requirements and technology constraints over the past decade have resulted in a radical rethinking of the processors, memories, and networks used in modern computing systems

Five Key Trends in Computer Architecture

1. Growing diversity in application requirements motivate growing diversity in computing systems pushing towards the cloud and IoT
2. Energy & power constrain systems across the computing spectrum
3. Transition to multiple cores integrated onto a single chip
4. Transition to heterogeneous systems-on-chip
5. Technology scaling challenges motivate new emerging compute, storage, and communication device technologies

Application

Algorithm

PL

OS

ISA

 μ Arch

RTL

Gates

Circuits

Devices

Technology

Agenda

What is Computer Architecture?

Activity 1

Trends in Computer Architecture

Activity 2

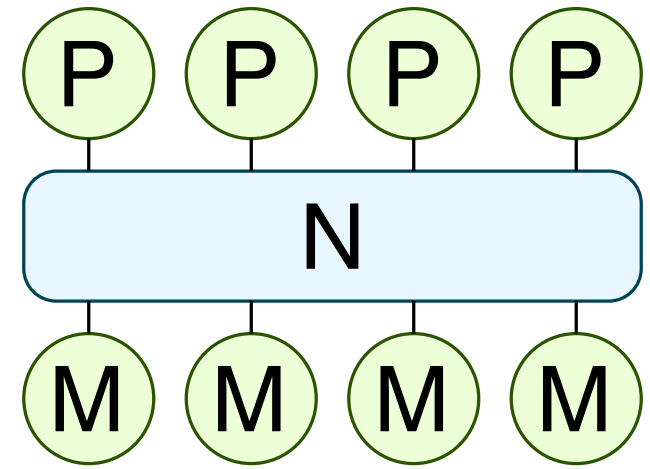
Computer Architecture Design

Activity #2: Sorting with a Parallel Processor

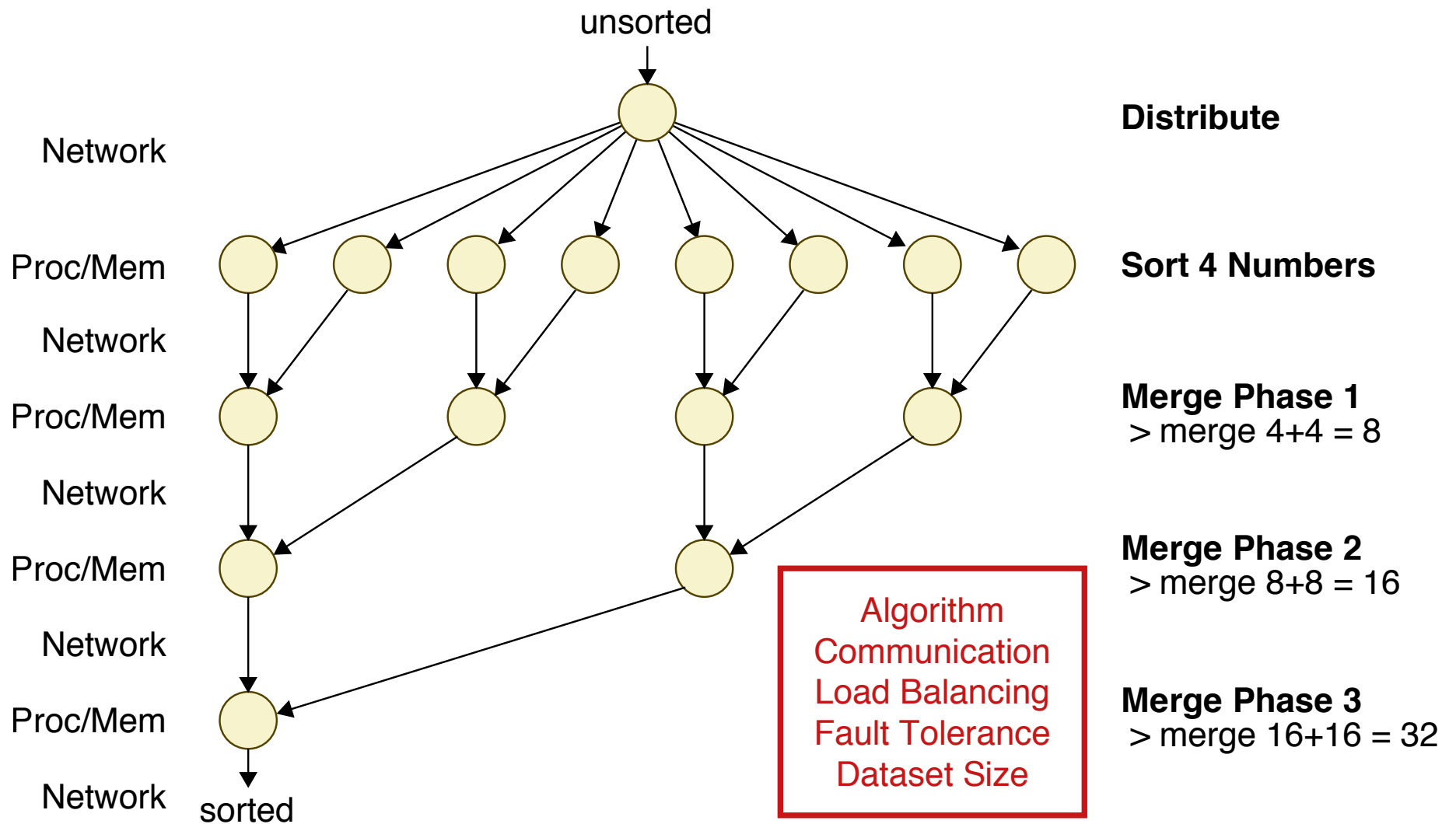
- ▶ **Application:** Sort 32 numbers
- ▶ **Simulated Parallel Computing System**
 - ▷ Processor: Group of 2–8 students
 - ▷ Memory: Worksheet, scratch paper
 - ▷ Network: Communicating between students

- ▶ **Activity Steps**

- ▷ 1. Discuss strategy with group
- ▷ 2. When instructor starts timer, master processor flips over worksheet
- ▷ 3. Sort 32 numbers as fast as possible
- ▷ 4. Lookup when completed and write time on worksheet
- ▷ 5. *Master processor only* raises hand
- ▷ 6. When everyone is finished, then analyze data



Activity #2: Discussion



Application

Algorithm

PL

OS

ISA

 μ Arch

RTL

Gates

Circuits

Devices

Technology

Agenda

What is Computer Architecture?

Activity 1

Trends in Computer Architecture

Activity 2

Computer Architecture Design

What do computer architects actually do?

General Science

Discover truths
about nature



Ask question
about nature

Construct
hypothesis

Test with
experiment

Analyze results and
draw conclusions

Computer Engineering

Explore design space
for a new system

Design and model
baseline system

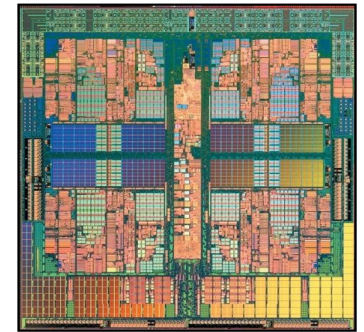
Ask question
about system

Test with
experiment

Analyze results and
draw conclusions

Build prototype
or real system

Design and model
alternative system



Modeling in Computer Architecture

Computer Engineering

Explore design space
for a new system

Design and model
baseline system

Ask question
about system

Test with
experiment

Analyze results and
draw conclusions

Build prototype
or real system

Design and model
alternative system

```
// rdy is OR of the AND of reqs and grants
assign in_rdy = | (reqs & grants);

reg [2:0] reqs;
always @(*) begin
    if ( in_val ) begin

        // eject packet if it is for this tile
        if ( dest == p_router_id )
            reqs = 3'b010;

        // otherwise, just pass it along ring
        else
            reqs = 3'b001;

    end else begin
        // if !val, don't request any ports
        reqs = 3'b000;
    end
end
```

Verilog • SystemVerilog • VHDL

C++ • SystemC

Bluespec • Chisel • Python

How do we design something so incredibly complex?

Computer Engineering

Explore design space
for a new system

Design and model
baseline system

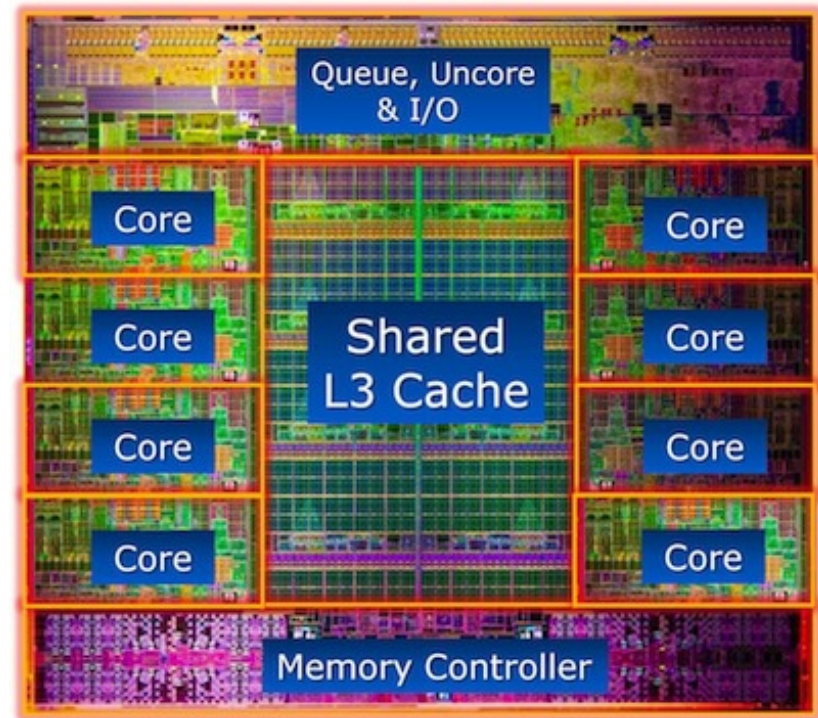
↓
Ask question
about system

↓
Test with
experiment

↓
Analyze results and
draw conclusions

↙
Build prototype
or real system

↘
Design and model
alternative system



Fighter Airplane: ~100,000 parts

Intel Sandy Bridge E:
2.27 Billion transistors

► Design Principles

- ▷ **Modularity** – Decompose into components with well-defined interfaces
- ▷ **Hierarchy** – Recursively apply modularity principle
- ▷ **Encapsulation** – Hide implementation details from interfaces
- ▷ **Regularity** – Leverage structure at various levels of abstraction
- ▷ **Extensibility** – Include mechanisms/hooks to simplify future changes

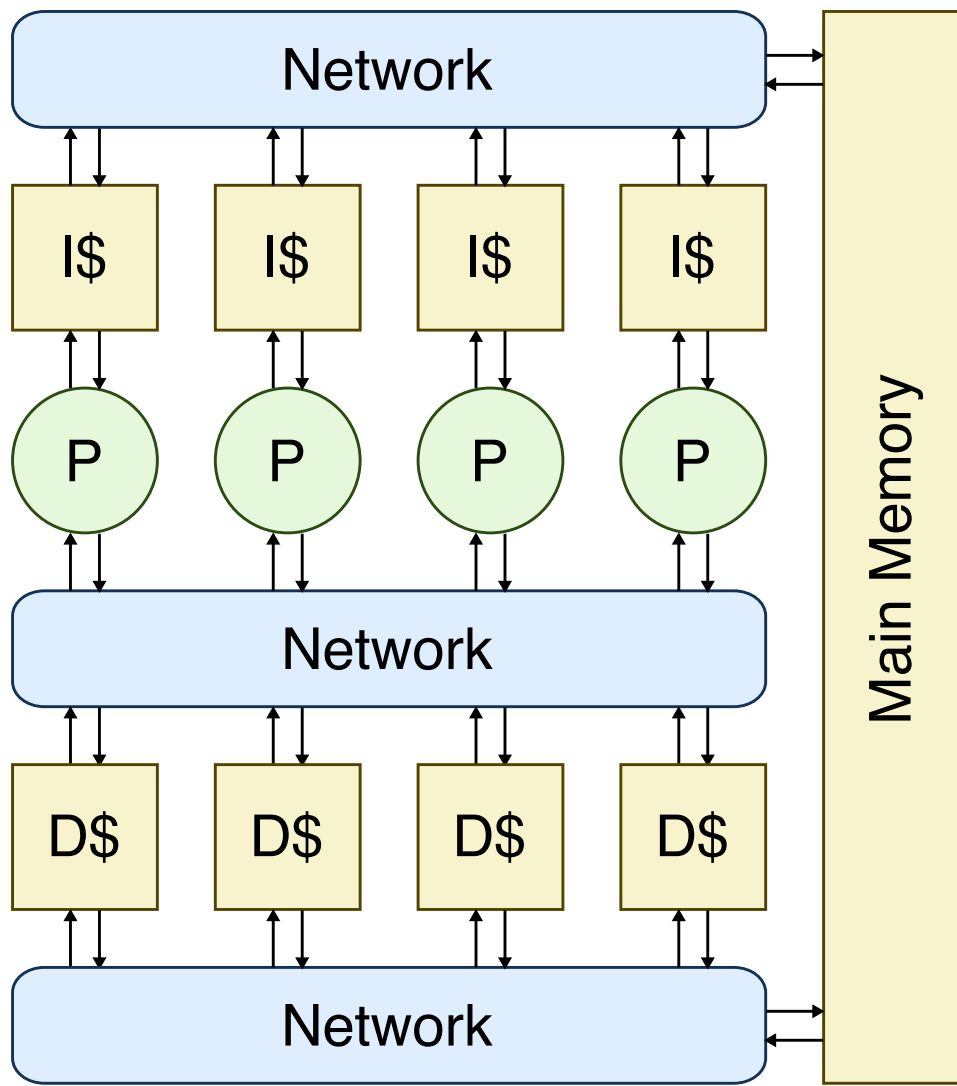
► Design Patterns

- ▷ Processors, Memories, Networks
- ▷ Control/Datapath Split
- ▷ Single-Cycle, FSM, Pipelined Control
- ▷ Raw Port, Message, Method Interfaces

► Design Methodologies

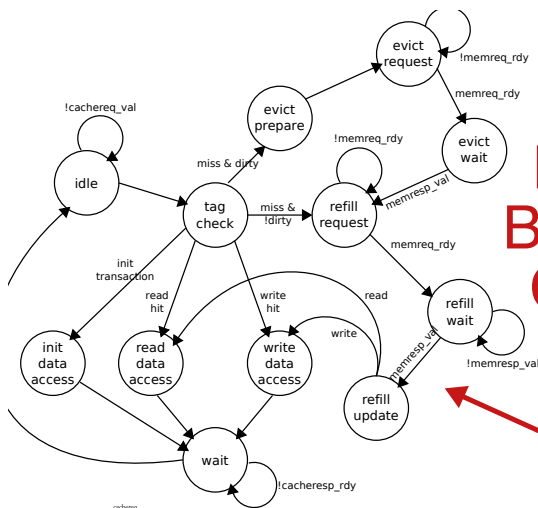
- ▷ Agile Hardware Development
- ▷ Test-driven Development
- ▷ Incremental Development

Final Goal for Lab Assignments

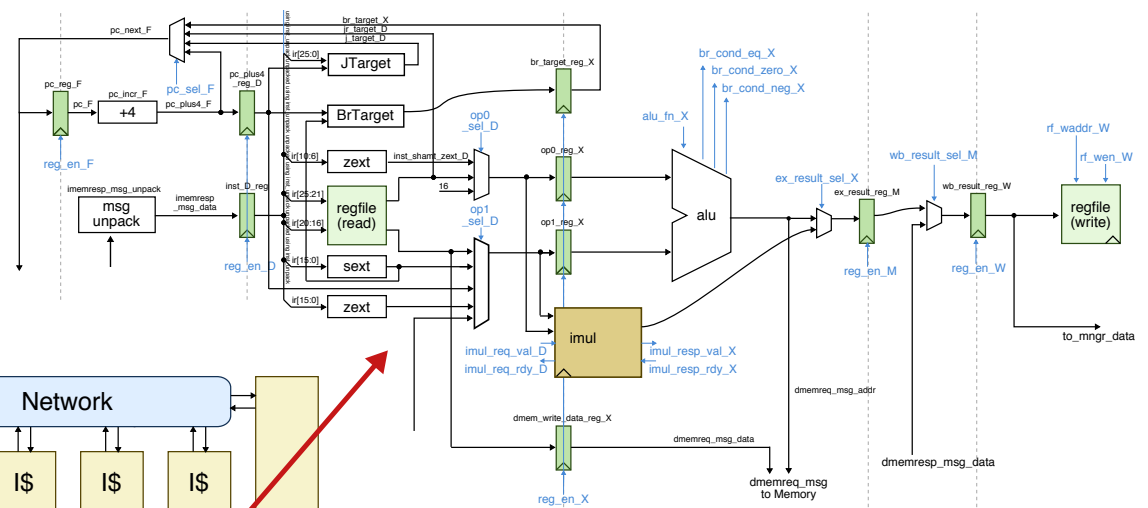


Quad-core processor with private L1 instruction caches and a shared, banked L1 data cache interconnected through various ring networks implemented at the register-transfer-level and capable running real parallel programs

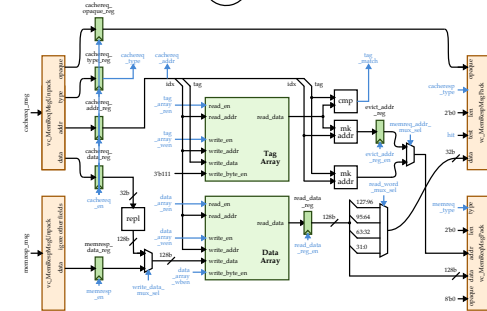
Lab assignments will use an agile hardware development methodology based on a **Python** hardware modeling framework, the **Verilog** hardware description language (optional), the **GitHub** repository hosting site, and the **TravisCI** continuous integration service



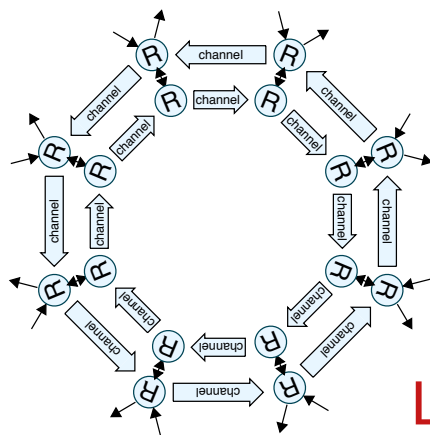
Lab 3: Blocking Cache



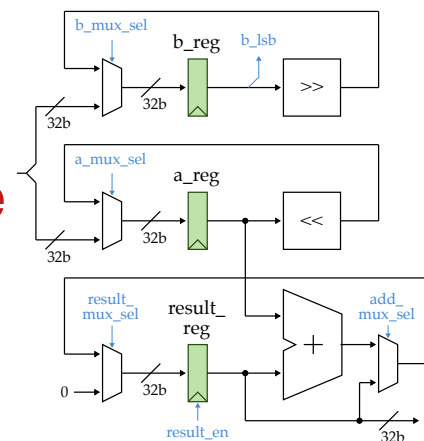
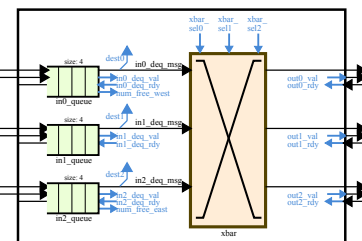
Lab 2: Pipelined Processor



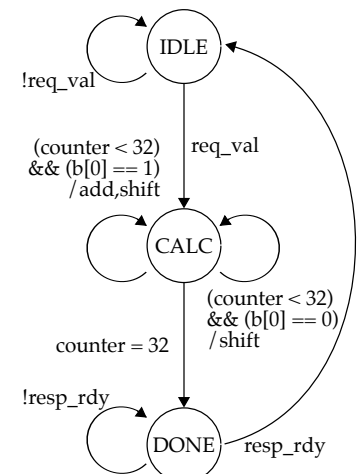
Lab 5: Multicore



Lab 4: Ring Network



Lab 1: Iterative Multiplier



Application

Algorithm

PL

OS

ISA

 μ Arch

RTL

Gates

Circuits

Devices

Technology

Take-Away Points

- ▶ Computer architecture is the process of building computing systems to meet given application requirements within physical technology constraints
- ▶ We are entering an **exciting new era of computer architecture** with growing diversity in applications and systems, a remarkable industrial shift towards mainstream parallel processing and SoCs, and significant technology scaling challenges
- ▶ This era offers tremendous challenges and opportunities, which makes it a **wonderful time to study and contribute to the field of computer architecture**